Practical 3: Challenge with SI Algorithms

Computational Intelligence Assessment: L7 SI

Zijin Hong
2020101911
Mathematics and Applied Mathematics
Jinan University - JBJI
hongzijin@stu2020.jnu.edu.cn

1 Introduction

This is a report of practical project[1]: Challenge with SI Algorithms, the catalogue including:

- PSO for solving N-Queen Problem.
- The corresponding required parts of the report.

2 Problem Description

The 16-Queens problem is a challenging variant of the classical N-Queens puzzle, which involves placing 16 queens on a 16x16 chessboard so that no two queens threaten each other. The challenge lies in ensuring that each queen is positioned such that no other queen shares the same row, column, or diagonal. This problem is significant in computer science as it encapsulates the complexity of constraint satisfaction and combinatorial problems, serving as a benchmark for testing various algorithms and heuristics.

In this report, we explore the PSO, traditionally used for continuous optimization problems, is adapted here to address the discrete nature of the 16-Queens puzzle. This adaptation and its effectiveness in finding a solution where all queens are placed on the board without attacking each other are the core focus of our study.

3 Solution Encoding

In the context of the 16-Queens problem solved via Particle Swarm Optimization (PSO), the solution encoding is a crucial aspect that bridges the gap between the continuous nature of PSO and the discrete structure of the problem. The encoding scheme represents how potential solutions (placements of queens on the chessboard) are represented within the algorithm.

3.1 Vector Representation:

The vector representation should be satisfied the following:

- Each solution is encoded as a 16-dimensional vector.
- Each element of the vector corresponds to a column on the chessboard, ranging from 0 to 15.
- The value of each element is an integer ranging from 0 to 15.

Example:

- A vector $[2, 4, 1, 3, \ldots]$ implies:

[1]The code is available in `https://github.com/Rcrossmeister/CI/tree/main/P3`

- The queen in the first column (0th index) is placed in row 2.
- The queen in the second column is placed in row 4, and so on.

This encoding scheme ensures that each queen is placed in a unique column. The challenge is to adjust their row positions such that they do not attack each other, adhering to the constraints of the 16-Queens problem. The discrete nature of the row positions in the vector must be carefully handled within the continuous framework of the PSO algorithm.

4 Objective Function and Constraint

4.1 Objective Function

The objective function in the PSO algorithm for the 16-Queens problem is designed to evaluate the quality of each solution, i.e., the placement of queens on the board. It calculates the number of pairs of queens that do not attack each other. The goal is to maximize this number, with the optimal value being when no queens are attacking each other.

The objective function is defined as:

$$\text{Fitness}(Q) = \text{Max Non-Attacking Pairs} - \text{Number of Attacking Pairs}(Q) \tag{1}$$

where Q represents the queen configuration in the solution vector.

4.2 Constraints

The constraints of the 16-Queens problem are inherent in the way queens can attack each other:

- No two queens can be in the same row.
- No two queens can be in the same column (implicitly satisfied by the solution encoding).
- No two queens can be on the same diagonal.

These constraints are respected by the fitness function, which penalizes configurations where queens attack each other. The optimal solution is achieved when the fitness function reaches its maximum value, indicating that no queens are attacking each other.

5 Models for PSO

In the application of Particle Swarm Optimization (PSO) to the 16-Queens problem, the standard model of PSO is adapted to suit the discrete nature of the problem. This subsection describes the key components and mechanisms of the PSO model used.

5.1 Particle Representation

In PSO, each particle represents a potential solution to the problem. For the 16-Queens problem:

- A particle is a 16-dimensional vector, with each dimension representing the position of a queen in a column on the chessboard.

5.2 Velocity and Position Update

The core of the PSO algorithm involves updating the velocity and position of each particle:

- **Velocity Update:** The velocity of each particle is adjusted at each iteration based on its own experience and the experience of its neighbors.
- **Position Update:** The position of each particle is updated by adding its velocity. Post-update, positions are discretized for compatibility with the chessboard grid.

5.3 Fitness Evaluation

The fitness of each particle is evaluated based on the objective function, which considers the number of non-attacking pairs of queens.

5.4 Algorithm Parameters

PSO involves several key parameters:

- **Inertia Weight (w):** Controls the impact of the previous velocity of a particle on its current velocity.
- **Cognitive Component (c1):** Represents the particle's cognitive influence, guiding it towards its personal best position.
- **Social Component (c2):** Represents the social influence, steering the particle towards the global best position.

These parameters are crucial in balancing exploration and exploitation in the search space, and their values are typically chosen based on empirical studies and problem-specific characteristics.

6 Algorithm Design with Pseudo Code and Parameters

The design of the Particle Swarm Optimization algorithm for the 16-Queens problem involves specific steps and parameters. The following pseudocode outlines the general procedure of the algorithm, highlighting the key components and operations.

6.1 Pseudo Code

```
1. Initialize a swarm of particles with random positions and velocities.
2. For each particle:
   a. Evaluate its fitness based on the objective function.
   b. Update the particle's personal best position if the current position
   is better.
3. Identify the global best position among all particles.
4. For each particle:
   a. Update velocity based on inertia weight, cognitive and social
   components.
   b. Update position by adding velocity to the current position.
   c. Apply discretization to map the continuous position to the
   chessboard grid.
5. Repeat steps 2-4 until a termination condition is met (e.g., solution
   found or maximum iterations reached).
6. Return the best solution found.
```

6.2 Parameters

The performance of the PSO algorithm depends on the following parameters:

- **Number of Particles:** The size of the swarm.
- **Inertia Weight (w):** Balances the exploration and exploitation abilities of the swarm.
- **Cognitive Coefficient (c1):** Determines how much particles are influenced by their personal best positions.
- **Social Coefficient (c2):** Determines how much particles are influenced by the global best position.
- **Maximum Iterations:** The maximum number of iterations before the algorithm terminates.

These parameters must be carefully tuned to ensure optimal performance of the PSO algorithm in solving the 16-Queens problem.

7 Result and Analysis

After executing the Particle Swarm Optimization algorithm for the 16-Queens problem, the following solution was obtained:

Solution:

```
[14  5  2 11 12  4  7  4  2 15  8  1  8  9  3 10]
```

Attacking pairs: 6

7.1 Analysis

The obtained solution indicates the row positions of the queens in each column of the chessboard. While the solution does not represent a completely non-attacking scenario (as the ideal number of attacking pairs is 0), it demonstrates a configuration where the number of attacking pairs is relatively low.

Key Points:

- **Non-Optimal Solution:** The presence of 6 attacking pairs suggests that the algorithm did not converge to an optimal solution. This could be due to the complexity of the problem or the need for further tuning of the PSO parameters.
- **Algorithm Performance:** The result highlights the challenges in adapting PSO, a method typically used for continuous optimization, to a discrete problem like the 16-Queens.
- **Parameter Tuning:** The outcome may be improved by adjusting parameters such as the number of particles, inertia weight, cognitive and social coefficients, or by increasing the number of iterations.

In conclusion, the PSO algorithm, with its current parameter settings, provided a near-optimal solution to the 16-Queens problem. Further research and experimentation with parameter tuning might lead to a fully optimal solution.

8 More Discussions

The application of Particle Swarm Optimization (PSO) to the 16-Queens problem presents several interesting discussion points:

8.1 Adaptation of PSO for Discrete Problems

PSO is traditionally used for continuous optimization problems. Its adaptation to a discrete problem like the 16-Queens puzzle involves challenges, especially in handling the discrete nature of chessboard positions within the continuous update framework of PSO.

8.2 Comparison with Other Algorithms

It would be insightful to compare the performance of PSO with other algorithms commonly used for the N-Queens problem, such as backtracking, genetic algorithms, or simulated annealing. Such a comparison could reveal strengths and limitations of PSO in handling discrete combinatorial problems.

8.3 Parameter Sensitivity

The performance of PSO is highly sensitive to its parameters, such as the number of particles, inertia weight, and cognitive and social coefficients. An extensive parameter tuning process or the use of adaptive parameter strategies could potentially enhance the algorithm's effectiveness.

8.4 Scalability

Another aspect worth exploring is the scalability of the PSO algorithm for larger instances of the N-Queens problem. Understanding how the algorithm performs as the size of the problem increases can provide insights into its suitability for large-scale combinatorial problems.

8.5 Hybrid Approaches

Combining PSO with other optimization techniques could lead to hybrid algorithms that leverage the strengths of multiple approaches. Such hybridizations might be more effective in finding optimal solutions to the N-Queens problem and similar puzzles.

8.6 Practical Implications

Finally, the study of algorithms like PSO in solving the N-Queens problem extends beyond theoretical interests. It offers practical insights into solving real-world problems that involve constraint satisfaction and combinatorial optimization, making such research valuable for complex decision-making scenarios.

In conclusion, the use of PSO for the 16-Queens problem opens up a wide array of research avenues, offering both challenges and opportunities for advancing the field of optimization and problem-solving.